Interim Guidance on
Systems Engineering Analysis
Required for ITS projects

By
Federal Highway Administration and
Maricopa Association of Governments

August 24, 2006

Introduction

A Systems Engineering Analysis is required for all Intelligent Transportation Systems (ITS) projects using Federal funds according to the Final Rule on Architecture and Standards Conformity[1]. The ITS Architecture Implementation Program identifies minimum systems engineering practices that must be included in the project implementation phase. The rule enacted by the U.S. Department of Transportation (DOT) on conformity with the National ITS Architecture requires regions to develop a regional ITS architecture. All subsequent ITS projects in the region must adhere to that regional ITS architecture. The regional ITS architecture for the MAG region was developed in 2001 and is documented in the MAG ITS Strategic Plan Update.

The goal of this Interim Guidance is to provide a better understanding of what steps should be included in a Systems Engineering Analysis for all federally funded ITS projects in the MAG region. The content of this document is based on previous work that is listed under References. This is an Interim Guidance for use by MAG member agencies pending the issuance of a guidance document to be released by the Federal Highway Administration. This guidance does not intend to impose a fixed structure for Systems Engineering Analysis. Rather, it simply provides general guidelines that member agencies can consult on. The process described here would be better termed a summary of the experience learnt by others in system development.

Systems Engineering Analysis can be defined as a structured process for planning, designing, validating, operating, maintaining and replacing/retiring a system. The process was first developed in the 1960s by the Department of Defense for hardware and software development. In fact, there is nothing new in Systems Engineering Analysis. Simply speaking, Systems Engineering Analysis generalizes the common steps in system development, highlights the key factors that need to be considered in most of the systems so that by conducting Systems Engineering Analysis, the system developers, managers and etc can reduce the risk of system failure. However, System Engineering Analysis also needs to be specified based on the particular system. A checklist may help to determine whether a certain factor or step is considered, but the answers, alternatives or methods to address this factor or step may vary. Even the same type of system could turn out to be quite different for different application environments. For example, a traffic signal system required for one city may not be the same as that in another. Another issue that must be mentioned is that the System Engineering Analysis steps and factors as stated in this document do not intend to provide a rigid framework that every ITS project development should include. Due to the diversity of the ITS projects, some steps/factors may not need to be considered in some projects; other steps/factors not mentioned in this document may be necessary in project development. The essence of the Systems Engineering Analysis is to provide a systems perspective during project development and to avoid missing key points that will affect the system during the project life cycle. There is no guarantee that a large, complex system would perform well by following the steps of Systems Engineering Analysis. However, it would have a high probability that the system would fail to perform at least some functions if not all without a Systems Engineering Analysis.

Process of Systems Engineering Analysis

The level of detail of a Systems Engineering Analysis would depend on the complexity and scale of a proposed system. It is important for the system stakeholders to appropriately evaluate the complexity of the system, the risk of system failure to determine the level of detail of the Systems Engineering Analysis. Usually, the more experience the system owners/stakeholders have, the more confidence they will have in evaluating system complexity and in controlling the level of details on Systems Engineering Analysis. On the other hand, the more uncertainty in the system life cycle, the more details will be required in the Systems Engineering Analysis to reduce the risk. While it is almost impossible for a Systems Engineering Analysis to address every single detail in a system's life cycle, it does make the stakeholders of a new ITS project pay attention to important details.

Many Systems Engineering Analysis models were developed in the past. One of the most popular models is the 'V' model. When applied to ITS projects, it is modified as shown in Figure1.

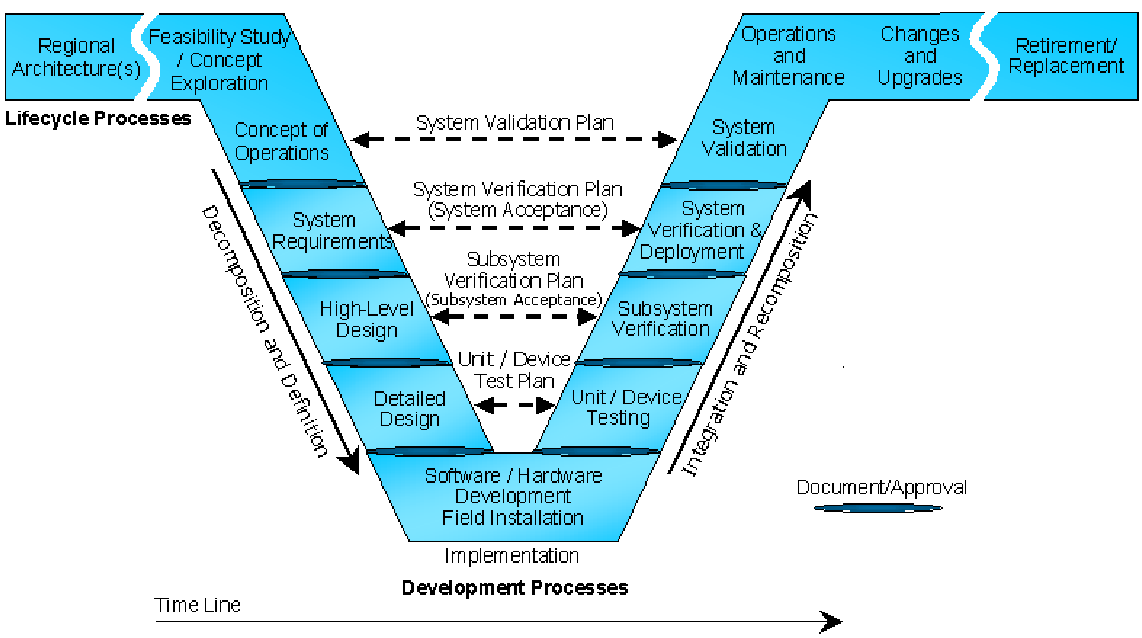

Figure 1 'V' model for Systems Engineering Analysis

The "V" model shows the early stages in building a system as steps along the left leg of the "V," the decomposition leg of the process. The steps on the decomposition leg break the system down into its pieces, proceeding from development of a Concept of Operations for the system, through the definition and refinement of the system's requirements (going from high-level to detailed requirements), to the system design stage, which also goes from high-level to detailed design. On the right-hand leg of the "V," we have the re-composition steps, where we take and test all of the parts of the system we've built and also put them together. As we proceed up the right-hand leg, we combine the system's building blocks into larger and larger pieces, until we have finally assembled and installed the complete system.

The "V" model also helps us understand the relationships between the work done on each side of the "V," as follows:

Conception (Concept of Operations stage) vs. Operations and Maintenance. During the Conception stage of the system life cycle, you create a *Concept of Operations* for the system. The Concept of Operations should describe how the system would work once it's built. Therefore, it relates directly to the Operation and Maintenance stage, during which the system's Concept of Operations is realized.

Looking at the explicit relationship between the two stages helps the systems engineer focus, at the beginning of the project, on issues associated with keeping the system in operation and effectively maintained once it's built. This long-range perspective is important in systems engineering.

System Requirements vs. System Acceptance. In the System Requirements Definition, you determine what the system has to do, when it has to do it, and how well it has to perform. In the System Acceptance stage, you determine whether the system you built satisfies those requirements. Satisfying those requirements involves two different approaches:

- Verification – ensuring that the desired functions have been implemented in the delivered system

- Validation – ensuring that all functions implemented in the system have been implemented correctly.

Another way of describing *verification* and *validation* is that verification is "building the right system" and validation is "building the system right."

System Design vs. Integration and Testing. The "V" diagram actually breaks this down into two components. System Design is subdivided into "High-Level Design" and "Detailed Design." Directly across from "High-Level Design" is "Subsystem Verification," which is a reasonable correlation. Part of what's done in the High-Level Design activity is to break the system down into its Subsystems, each of which is assigned a major functional area of the overall system. In Subsystem Verification, you're integrating all of the components of the subsystem and testing the Subsystem(s) as units. It's the appropriate way to look at how the two stages are correlated.

"Detailed Design" is directly across from "Unit/Device Testing". Detailed Design relates to the definition of the complete system, the final design of all of the elements, at the level necessary to build it bottom-up. The Unit/Device Testing portion of the "V" is the start up on the leg where you bring all of the pieces together. During this stage, you'll test individual components as "units" and begin testing the individual units that interact with one another, preparing to fit them all together into individual subsystems.

The design of a system, in part, allocates functions to be performed by specific system components. In addition, during design, we decide how we're going to implement those functions. That may involve developing *algorithms* (i.e., specific, detailed instructions on how to perform a function) or developing *interfaces* that allow devices in our system to interoperate. The integration and testing of a system involves ensuring, at all levels of the system, that each piece works as it should and that all pieces successfully and accurately interact.

The design stage is the final *decomposition* stage of the decomposition leg of system development. It's the stage where you establish, at the most detailed level, your plan for how you will accomplish the work of the system. The Unit/Device Testing stage is the first of the major *re-composition* stages, where you begin assembling the pieces of the system into an integrated whole.

At the bottom of the "V" is the *Implementation* stage. This stage represents the transition from decomposition (the conceptual level) to re-composition (the physical level). During this stage, we transform the system's design into actual products.

A quick aside on when certain things are done in a systems project. It's fairly common on ITS projects for public sector agencies to hire contractors either to build the ITS system or to integrate it using commercial off-the-shelf (COTS) components, along with any custom-built hardware or software needed to tie everything together. It's important for the public sector to work together with its contractors in the requirements definition, analysis, and refinement process. One reason is that the system contractors usually have more experience in developing good requirements than do public sector agencies. Developing sound requirements is a major part of the contractors' business. Thus, they should have the tools and skills to take the high-level user requirements and convert them into a solid basis for a buildable system.

A second reason for the public sector to work together with system contractors on requirements is the need for everyone on the project to have a common understanding of what each requirement means. Writing a requirement down doesn't ensure that everyone will interpret it the same way. English is an imperfect language for documenting requirements. The writer and reader of a requirement can legitimately, honestly, and with no mal-intent differ on what it means. If the public sector and the system developer have different interpretations of any requirements and these different interpretations are not resolved early in the project, the misunderstandings can have serious negative impact on the project.

One way to achieve a common understanding between the public sector and the system developer on the meaning of requirements is to conduct a requirements review. This involves having all parties review requirements together, discuss them, and agree on their meaning. A requirements review can be a lengthy, painstaking process; field experience on ITS projects indicates that it's a valuable one.

Steps of the Systems Engineering Analysis as depicted in the 'V' model are discussed in the following sections:

Step 1: Interfacing with the Regional ITS Architecture

This step would clarify what particular transportation needs would be met by the project and what components in the Regional ITS Architecture are included in it. Meanwhile, it also establishes a clear map connecting the vision (Regional ITS Architecture) and the paths (individual ITS projects) to achieve the vision.

In general, this step links the project with the Regional ITS Architecture (RIA). It would identify the relevant ITS Subsystems, User Needs, ITS User Services, Market Packages and Architecture Flows in the Regional ITS Architecture that are covered by the project. If some of these project attributes are not included in the RIA, it is recommended to make arrangements to update the Regional ITS Architecture.

Usually, a clear statement about project goals, objectives and description will provide most of the information on this step. It helps the system developers to answer the question of what to do.

Step 2: Feasibility Study

The feasibility of the project is explored in this step. Usually, at least three aspects will be addressed: the technical feasibility, financial feasibility and the institutional feasibility. The detail level of the feasibility study depends on the complexity and scope of the project. The question of 'can it be done' is answered in this step.

Step 3: Project Planning and Concept of Operations

In project planning, all relevant agency policies and procedures on managing and executing such a project are identified. The role and responsibility of each stakeholder should also be clarified. Then the project tasks (both administrative and technical), their interdependencies, estimates of needed resources and budget for each task, the project schedule and the project's risks are also identified.

Concept of Operations envisions how the system is to operate and how the system will meet the needs and expectations of the stakeholders. The envisioned operation is defined from multiple viewpoints of stakeholders such as operators, maintainers, and managers. The Concept of Operations also serves as the basis for developing the system validation plan. The system validation plan will address how the system is validated (proof that the envisioned system meets the intended needs).

Step 4: System Requirements Definition

This step starts with high-level requirements, usually derived from the Concept of Operations for the system and then refines and expands on these requirements descriptions to produce a sound set of Functional Requirements. System Requirements need to be defined at two-levels:(1) high-level User Service Requirements and (2) more detailed low-level Functional Requirements. The high-level User Service Requirements should be developed based on the first three steps and presented in the form of "shall" statements. An example for a User Service Requirement is -- a project shall support real-time video image transfer between two TMCs. Once the high-level

User Service Requirements are defined, they are detailed further into low-level Functional Requirements. For example, ‘the video images shall be updated every 1 minute’ is a lower level Functional Requirement for the above stated User Service Requirement. With the Functional Requirements defined, alternative communications technologies that meet the requirements can be selected for Alternatives Analysis. System Requirements Definition generates information that would be used later to verify if the system would perform as expected. As a result, System and Subsystem Verification Plans should be developed following the Requirements Definition.

Step 5: System Design

Similar to the Requirements Definition, System Design involves two levels of design: (1) High-Level Design and (2) Detailed Design at the lower or component level. While the High-Level Design focuses on project architecture such as interfaces among system components and Alternatives Analysis to compare different technologies, the Detailed Design addresses how the system components would meet the Functional Requirements as well as how the components work together to accomplish the overall system goals and objectives. The low-level Detailed Design should produce a design document with unique explanations. That is, regardless of who builds the system, there should not be significant differences in the final system. System Design is a bridge between vision and reality. Careful design review is necessary. Component and unit level verification plans should be developed.

Step 6: System Implementation

This step involves a review of the planned procurement process, hardware fabrication, software coding, database implementation, configuration of off-the-shelf products and field installation. When most of work in this step is planned to be conducted by consultants, system owners and stakeholders should plan for periodic reviews to ensure project deployment and to solve unexpected situations.

Step 7: System Test and Verification

This step involves the development of plans for testing, verifying and validating the functions of the new ITS project. By following these plans, system developers can make sure if the constructed system performs as envisioned. Moreover, they can help identify bugs and errors in the system. Contrary to the system planning stage, where the system is decomposed into Subsystems, Components, even Units, the System Test and Verification step is a composition process. Because it would be much easier to identify problems in small simple units than large systems, the System Test and Verification should start with each single unit. Once the problems in each unit are identified and fixed, the Test and Verification proceeds to the next level until the entire system as a whole is tested and validated. Only when each unit, component, subsystem and the entire system is verified and validated, can the project be accepted.

Step 8: System Operation and Maintenance

This step would address system operation and maintenance requirements by identifying required funding, staff and other resources for the project. In addition, it will document how the system is expected to be operated and maintained, as they are also important for ensuring that the system performs as planned. The Requirements Definition would have addressed and defined what is required for the man-machine interaction. An operation and maintenance manual detailing the operation and maintenance standards and procedures will greatly reduce human errors as well as increasing system reliability.

Step 9: System Update, Retirement and Replacement

With the development in technologies and any changes of the user needs, a system may become obsolete and need to be altered, upgraded or replaced. Strategy plans on how to update, retire or replace the system needs to be developed to reduce the impact during system change. Attentions must be paid on the compatibility of the old system with the new parts.

Summary of Products in A System Engineering Analysis

1. A clear statement about project goals, objectives and description to connect the ITS project with the regional ITS architecture.
2. Feasibility study to answer if the project goals, objectives can be realized given the technical, financial, institutional constraints.
3. Project plan to address responsibility of the stakeholders, major project tasks, project schedule and the Concept of Operations.
4. A definition of high-level User Service Requirements and low-level Functional Requirements to translate project goals and objectives into practical requirements for the ITS project. These requirements will also serve as basis for system design.
5. System and Subsystem level verification plan based on Requirements Definition, including procedures to verify if a subsystem or system performs as stated in the requirements.
6. Alternatives studies to select the most appropriate technologies, procurement options, ITS standards and etc. including cost-benefit analysis.
7. Plan for design reviews to ensure the design meets the proposed requirements
8. Component and unit level verification plan based on design, including procedures to verify if a component or unit perform as stated in the design document.
9. Project implementation plan to install the system as design, including periodic reviews of the implementation.
10. Develop procedures to conduct Unit, Component, Subsystem and System level verification and validation.
11. Plan to identify operation and maintenance resources, responsibilities and standardize system operation and maintenance procedure.
12. Alternative studies for system upgrade and strategy plan for updating, retiring and replacing the system.

For highly complex projects, other products such as a Risk Management Plan may also be included in the Systems Engineering Analysis.

References:

1. 23 CFR Parts 655 and 940, Intelligent Transportation Systems Architecture and Standards
2. Guidance on Using the Systems Engineering Process in ITS Projects, Alan Hansen and James Colyar, FHWA Arizona Division Office, August 2002
3. Caltrans System Engineering Process Checklist
4. Caltrans System Engineering Guide Book for ITS
5. Developing Functional Requirements for ITS Projects, ITS JPO, USDOT, March 2002 http://www.itsdocs.fhwa.dot.gov/JPODOCS/REPTS_TE/13621.html